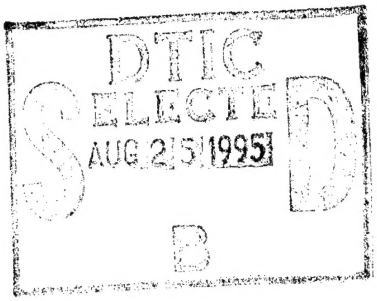
NAVAL POSTGRADUATE SCHOOL
Monterey, California



THESIS

ANALYSIS OF THE DES, LOKI, AND IDEA ALGORITHMS FOR USE IN AN ENCRYPTED VOICE PC NETWORK

by

Walter O. McClenney

March 1995

Principal Advisor: Chin-Hwa Lee

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 1995	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE Analysis of the DES, LOKI, and IDEA Algorithms for use in an Encrypted Voice PC Network

5. FUNDING NUMBERS

6. AUTHOR(S) McClenney, Walter O.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)* The protection of information is of vital importance to the successful operation of both the Federal Government and the Department of the Navy. Attention is usually given to the protection of classified information. The Computer Security Act mandates that not only classified, but sensitive information be protected in accordance with the Privacy Act. The increasing reliance on networks makes this a challenging problem to overcome.

The focus of this thesis is to examine the capabilities, effectiveness, and limitations of the DES, LOKI, and IDEA cryptosystems for use in the PC environment. It analyzes the use of these cryptosystems for network voice encryption. Further, this thesis examines the function and security of these cryptosystems and on possibilities for implementation in a Naval Postgraduate School PC network.

Experimental results on the speed and efficiency of two of the cryptosystems are presented to show their relative strengths and weaknesses. A recommendation is made as to the appropriate cryptosystem to be used in a network implementation. Further recommendations are given on the type of computer networking architecture and the type of network encryption to use.

14. SUBJECT TERMS Networking, Encryption, Network Audio	15. NUMBER OF PAGES 58
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited.

ANALYSIS OF THE DES, LOKI, AND IDEA ALGORITHMS FOR USE IN AN ENCRYPTED VOICE PC NETWORK

Walter O. McClenney
Lieutenant, United States Navy
B.A., University of Virginia, 1987

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
March, 1995

Author: _____
Walter O. McClenney

Approved by: _____
Chin-Hwa Lee, Principal Advisor

_____
Myung W. Suh, Associate Advisor

_____
David R. Whipple, Chairman
Department of Systems Management

ABSTRACT

The protection of information is of vital importance to the successful operation of both the Federal Government and the Department of the Navy. Attention is usually given to the protection of classified information. The Computer Security Act mandates that not only classified, but sensitive information be protected in accordance with the Privacy Act. The increasing reliance on networks makes this a challenging problem to overcome.

The focus of this thesis is to examine the capabilities, effectiveness, and limitations of the DES, LOKI, and IDEA cryptosystems for use in the PC environment. It analyzes the use of these cryptosystems for network voice encryption. Further, this thesis examines the function and security of these cryptosystems and examines possibilities for implementation in a Naval Postgraduate School PC network.

Experimental results on the speed and efficiency of two of the cryptosystems are presented to show their relative strengths and weaknesses. A recommendation is made as to the appropriate cryptosystem to be used in a network implementation. Further recommendations are given on the type of computer networking architecture and the type of network encryption to use.

TABLE OF CONTENTS

I. INTRODUCTION

A. PURPOSE

The information age has brought an increasing reliance on computers, communications, and the networks that support them. With the increasing reliance on computer and communications networks, the need for security tools to protect the information carried on these networks becomes evident. The Department of Defense (DoD) and the Department of the Navy (DoN) have special concerns in the information security arena. Both DoD and DoN have not only large volumes of classified information but large volumes of sensitive information which must be protected as well.

As the available bandwidth increases in the ever evolving networks, many different types of applications are being used. Video and audio in network applications are becoming more and more common. This thesis looks specifically at network voice applications and software encryption methods for protecting that information. The intent is to present an overview of three of the current cryptosystems and give possible solutions for using one of those cryptosystems in an encrypted voice PC network.

B. OBJECTIVES

This thesis examines the capabilities, effectiveness, and limitations of LOKI, DES and IDEA cryptosystems for use in the PC environment. In particular it analyzes the use of these cryptosystems for network voice file encryption. The paper begins with a thorough review of the background concepts of encryption and networking. Included in the background information are the basic concepts of cryptography and an introduction to the three cryptosystems studied in this thesis. The background information also covers a comprehensive overview of networking, including a description of the two major networking architectures.

Following the background information, the thesis provides a detailed analysis of the DES, LOKI and IDEA encryption algorithms. Both the function and block diagram of each algorithm is presented. A brief examination of the current state of cryptanalysis is

presented for the three algorithms. The various modes of operation of the cryptosystems are presented as well.

A thorough understanding of the background material is necessary to make a proposal for implementation in a PC network. Considerable research has been done involving the increasing use of computer networks. Computer security is still one of the weak areas of both industry and DoD. This thesis addresses the networking issue from the standpoint of PC voice implementation and the security issue from the standpoint of encrypting that voice traffic.

Following the presentation of the DES, LOKI and IDEA cryptosystems, experimental results using these cryptosystems are provided. A proposal for implementing an encrypted voice PC network will also be provided. It is hoped that further research will be devoted to expanding the implementation portion of this thesis.

C. RESEARCH QUESTIONS

1. Primary Research Question

- What are the capabilities, effectiveness, and limitations of LOKI, Data Encryption Standard (DES) and International Data Encryption Algorithm (IDEA) cryptosystems?

2. Secondary Research Questions

- What are the major areas of concern of encryption in network configuration?

- What are the possibilities for implementing an encrypted PC voice network in a Naval Postgraduate School?

D. SCOPE OF THE THESIS

This thesis studies the strengths and weaknesses of the DES, LOKI, and IDEA cryptosystems. Research was conducted on the functional workings of these cryptosystems, and the relative strengths and weaknesses of each. Further, this thesis investigates the requirements for implementing one of these cryptosystems in a PC

network. In making a proposal for implementation, research was directed toward finding low-cost options for implementing the network.

E. LITERATURE REVIEW AND RESEARCH METHODOLOGY

The overall research is a review of archival-based publications, reports. and studies on this subject. There was significant use of the Internet to obtain the collected documents and to conduct one-to-one correspondence with experts in the field.

Networking publications were studied in this research in order to identify common problem areas for this type of network implementation. In addition to the networking periodicals, several encryption software applications readily available in the shareware or public domain were reviewed in order to identify similar problems in cryptography software implementation. Several networking applications were considered to determine a possible candidate for implementation.

Several experiments were performed as a means of comparing the speed and efficiency of software enryption programs. In all of the experiments, "wall time" was used in order to measure the speed. The programs were run on a 386/33mHz IBM compatible PC.

F. ORGANIZATION

This thesis is organized in five chapters. The remainder of this thesis is organized as follows:

- Chapter II, "Background," provides a general background on networks and encryption.

- Chapter III, "DES, IDEA, and LOKI Cryptosystem Descriptions," provides an in-depth look at the functionality, strengths, and weaknesses of these cryptosystems.

- Chapter IV, "An Implementation Proposal," presents experimental results on the cryptosystems and provides a possible low cost solution for implementing an encrypted voice PC network.

- Chapter V, "Conclusions," identifies the overall conclusions of this research.

II. BACKGROUND

A. INTRODUCTION

The protection of information has always been vital to the successful operation of both the Federal Government and the Department of the Navy. Considerable attention is usually given to the protection of classified information. This paper will review software encryption for the protection of information which is unclassified , but which may be considered sensitive. The Computer Security Act (CSA) gives broad definition to the term "sensitive information":

> ...information whose loss, misuse, unauthorized access to, or modification of could adversely affect the national interest, or the conduct of federal programs, or the privacy to which individuals are entitled to under the Privacy Act [Ref. 23].

This would include information such as an individuals social security number or other personal information. The CSA also requires every U. S. government computer system that processes such sensitive information to have a customized security plan. This thesis will hopefully give some insight into one way of helping to protect that information, through the use of encryption of voice traffic on a PC network.

Any study of network encryption must begin with an overview of the practice of, and terminology associated with cryptography, as well as an overview of network practice and terminology. The focus of this chapter is to provide the relevant background material for this work. Initially this chapter will provide definitions for some of the terminology to be used in this paper. Secondly, a review of the basics of cryptography will be provided. Finally, a review of basic networking and network encryption principles will be discussed.

B. DEFINITION OF RELEVANT TERMS

It is important that the reader have a clear understanding of certain key terms used throughout this paper. The key terminology listed both here and in Appendix D is used frequently in the professional realm of both networking and cryptography. The key terms are:

- **OSI reference model** - A set of international standards that provides a common set of conventions for computer communications and networking. The OSI reference model provides a framework for defining standards for linking heterogeneous computers. The model was designed to make each layer manageably small, but not have so many layers as to become burdensome. The layers include: physical, data link, network, transport, session, presentation, application [Ref. 21].

- **TCP/IP**- A view of communication that uses three proxies,: processes, hosts and networks. These three concepts yield to a fundamental principle of the TCP/IP protocol suite: the transfer of information to a process can be accomplished by first getting it to the host in which the process resides and then getting it to the process within the host [Ref. 21].

- **Link-to-Link Encryption**- Devices connected at the physical layer encrypt all data passing through them, including data, routing information, protocol information, etc. Link-to-Link is the simplest way to add encryption in a network [Schneier, p.178]. It is performed by the lowest two levels in the OSI stack. The lower level layers are the most standardized levels. It is invisible to the user. The messages are only encrypted in transit and are plaintext on host [Ref. 22].

- **End-to-end encryption**- Put encryption equipment between the network layer and the transport layer. The encryption device must understand the data according to the protocols up to layer three and encrypt only the transport data units. Performed by high level protocol layers. The user must initiate the encryption process. The data is encrypted throughout the network [Ref. 22].

- **Keys**- All modern encryption algorithms use a key which can take one of many values (a large number is best) [Ref. 19].

- **Asymmetric algorithms**- Algorithms which use separate encryption and decryption keys, public-key algorithms are examples of asymmetric algorithms [Ref. 19].

- **Symmetric algorithms**- Algorithms which use the same key for encryption and decryption [Ref. 19].

- **Stream ciphers**- Algorithms which encrypt information bit-by bit [Ref. 19].

- **Block ciphers**- Algorithms which encrypt information in groups of bits typically 64 bits [Ref. 19].

C. OVERVIEW OF CRYPTOGRAPHY

Cryptography is the science and art of keeping messages secure. Messages are encrypted so that only the sender and the intended recipient are able to read them. These messages may occur in a variety of forms: written text, stored as files on floppies, or sent from computer to computer through a network. The general strategy behind any cryptographic algorithm is to take *plaintext* P put it through an *encryption function* E and create *ciphertext* C using the form:

$$C=E(P)$$

The recipient of C must then be able to take the encrypted message and using a decryption function D turn C back into P using the form:

$$P=D(C)$$

Since the whole point of encrypting and then decrypting a message is to recover the original plaintext, the following identity must hold true :

$$P=D(E(P))$$

Cryptographic algorithms are the mathematical functions used to carry out encryption and decryption. There are two general types of cryptographic algorithms, symmetric and asymmetric. Symmetric algorithms can be divided into two different categories: stream ciphers and block ciphers. The ciphers discussed in this paper, Data

Encryption Standard (DES), IDEA, and LOKI are all examples of symmetric block cipher systems.

Block algorithms achieve their cryptographic effect through *confusion* and *substitution*. *Confusion* hides the similarities between the plaintext and the ciphertext. This is most commonly done through *substitution* [Ref. 19]. In substitution, a letter or block of plaintext is replaced with a different letter or block of ciphertext. Looking for repetition in the ciphertext is one of the ways to "break" an encrypted message. Diffusion dissipates the redundancy of the plaintext by spreading it over the ciphertext [Ref. 19]. Diffusion is most commonly done through the use of permutation. Algorithms may also use more complicated methods, some of which will be described later in this thesis.

The Data Encryption Standard (DES) is a block cipher, symmetric key cryptosystem incorporating both transposition and substitution. The algorithm requires a 64-bit key (only 56-bits are actually used). DES encrypts data in 64-bit blocks, taking a 64-bit block of plaintext and converting it into a 64-bit ciphertext block. The algorithm uses only standard arithmetic and logical operations on up to 64-bit numbers. It is easily implemented in either hardware or software.

The IDEA algorithm is a block cipher, symmetric cryptosystem incorporating both confusion and substitution. The algorithm uses a 128-bit key. IDEA encrypts data in 64-bit blocks, taking a 64-bit block of plaintext, breaking it up into 4, 16-bit blocks to employ the algorithm. The design philosophy behind the algorithm is one of mixing operations from different algebraic groups [Ref. 19]. The algorithm is easily implemented on 16-bit processors.

The LOKI algorithm is a block cipher, symmetric cryptosystem which operates very similarly to DES. The algorithm uses a 64-bit key. LOKI encrypts data in 64-bit blocks, taking a 64-bit block of plaintext, breaking it up into 2, 32-bit blocks to employ the algorithm. LOKI uses many of the same arithmetic and logical operators as DES, except unlike DES there is no initial permutation, the data block is first XORed with the key. Like DES, the algorithm is relatively easy to implement in software.

The relative merits of symmetric algorithms versus asymmetric algorithms can be debated ad nauseam. However, there are certain factors which must be taken into consideration. The implementation of the encryption algorithm can be in either hardware or software. This paper will look at implementing the algorithm in software. Software implementation requires speed to be a prime factor when considering an encryption schema. A disadvantage of using asymmetric algorithms is speed, therefore symmetric algorithms were used [Ref. 9]. Key distribution is another important factor when considering the choice of a cryptosystem. Symmetric key algorithms require both users to share and maintain the secrecy of the key. Asymmetric key algorithms have the advantage of each user being responsible for his own key, however key distribution of the public key is still an issue.

Cryptanalysis is the attempt to "break" a cryptosystem. All cryptosystems have certain properties which they must satisfy. Cryptosystems which have not been subjected to cryptanalysis should be viewed with skepticism. The security of symmetric cryptosystems, such as the ones reviewed in this work, is a function of two things: the strength of the algorithm and the length of the key [Ref. 19].

> The latter is easier to demonstrate....Assume that the strength of the algorithm is perfect....By perfect, I mean that there is no better way to break the cryptosystem other than to try every possible key; this is called a brute-force attack [or exhaustive key search]. If the key is 8 bits long, then there are 2^8=256 possible keys. Therefore it will take 256 attempts to find the correct key, with an expected number of attempts of 128. If the key is 56 bits long, then there are 2^{56} possible keys. Assuming a supercomputer can try a million keys a second, it will take 2000 years to find the correct key. [Ref. 19]

Properties which all cryptosystems should exhibit are:

- The security of the cryptosystem rests with the secrecy of the key rather than with the supposed secrecy of the algorithm [Ref. 19].

- A strong cryptosystem has a large keyspace [Ref. 19].

- A strong cryptosystem will produce ciphertext which appears random to all standard statistical tests [Ref. 9].

- A strong cryptosystem will resist all known previous attacks [Ref. 9].

Cryptosystems which meet the above tests are not guaranteed to be secure. Time is the true test of a good cryptosystem. This paper will review the status of cryptanalysis of IDEA, DES, and LOKI cryptosystems.

D. OVERVIEW OF NETWORKS

Communications techniques cover a full spectrum of systems from sending a letter through the Postal Service to information exchange of complex client-server nodes networks. In its simplest form, information communication takes place between two devices that are directly connected by some form of transmission channel. Often, however it is impractical for two devices to be directly connected [Ref. 21]. More and more, communications and computers are merging into one entity via the connection of networks. In the advent of this fact, several key factors have emerged:

- There is no fundamental difference between data processing (computers) and data communications (transmission and switching equipment) [Ref. 21].

- There are no fundamental differences among data, voice, and video communications via digital techniques [Ref. 21].

- The lines between single-processor computer, multi-processor computer, local network, metropolitan network, and long-haul network have blurred. [Ref. 21]

Network communications have developed into an indispensable tool both in industry and DoD. There are numerous type of communications and computer networks to deal with. There are some things however that all networks have in common in order to communicate with each other. In discussing computer communications and computer networks, two concepts are paramount: protocols, which are conventions for the

exchange of data between two entities, and computer-communications architecture, the collection of modules that implements the communications functions [Ref. 21].

1. Protocols

Most systems are implemented with layered protocol architecture such as OSI or TCP/IP [Ref. 3] The key elements of a protocol are:

- Syntax: includes such things as data format, coding, and signal levels [Ref. 21].
- Timing: includes speed matching and sequencing [Ref. 21].
- Semantics: includes control information for coordination and error handling [Ref. 21].

Along with these key elements, there are several important characteristics of protocols:

- Direct/indirect [Ref. 21].
- Monolithic/structured [Ref. 21].
- Symmetric/asymmetric [Ref. 21].
- Standard/nonstandard [Ref. 21].

Communications between two points may be either direct or indirect. If the two points share a direct link then they communicate directly. It is of course possible for there to be more than two pints in the communications link and still have direct communications. If a pair of communicating devices go through some kind of arbitrated or third party in order to communicate then that is an indirect link.

A monolithic protocol is one in which all of the elements of a single protocol are reflected in all communicating devices. This makes communicating difficult if not

impossible for networks with differing monolithic protocols. All of the processes must have the protocol logic as well.

> ...Consider an electronic mail package running on two computers connected by a synchronous HDLC link. To be truly monolithic, the package would need to include all of the HDLC logic. If the connection were over a packet-switched network, the package would still need the HDLC logic (or some equivalent) to attach to the network [Ref. 21].

Structured protocols are a result of structured design and implementation principles [Ref. 21]. In order to reduce the complexities which would be involved with a monolithic protocol, protocols can be layered. Instead of a single protocol, the protocols can be arranged in a hierarchical structure [Ref. 21].

Lower-level, more primitive functions are implemented in lower-level entities that provide services to higher-level entities. For example, there could be an HDLC module (entity) that is invoked by an electronic mail facility when needed [Ref. 21].

Symmetry and asymmetry involve the level at which the communication is being carried out. Communications which are between similar elements, i.e. two user PC's on a network, are symmetric.

> Asymmetry may be dictated by the logic of the exchange (e.g., a "user" and a "server" process), or by the desire to keep one of the entities or systems as simple as possible. An example would be the normal response mode of HDLC [Ref. 21].

Standard protocols refer to protocols which have applicability over a wide range of platforms. Standard protocols have usually been adopted by some type of organizing body, and generally have wide acceptance both on the part of industry and users. Non-standard protocols exist as "stovepipe" systems. They are generally designed for specific applications and for specific platforms. Both industry and DoD have moved away from non-standard protocols because of the expense involved and the lack of portability with other non-standard systems.

2. Computer Networking Architecture

It is possible to look at the computer architecture as a structured collection of protocols which facilitate the exchange of data over a computer or communications network. It is a set of policies which will lead to better overall connectivity [Ref. 20]. OSI, TCP/IP, and SNA are all examples of computer architectures. This paper will review the basic functions of the OSI, and TCP/IP architecture.

a. OSI Architecture

The Open Systems Interconnection (OSI) model is a set of international standards that provides a common set of forms for computer communications and networking. The OSI reference model provides a framework for defining standards for linking heterogeneous computers. The model was designed to make each layer manageably small, but not have so many layers as to become burdensome [Ref. 21]. The layers include: physical, data link, network, transport, session, presentation, application. The strength of the OSI is that it is specifically designed for heterogeneous communications. The protocol layers communicate on a peer-to-peer basis, making the communications issues less complex by breaking them up into functional units.

Each of the seven layers of the OSI model has a specific task to perform in the architecture. From top to bottom the functions of the layers are:

- Application - contains management functions and generally useful mechanisms to support distributed applications. Examples are file transfer and electronic mail [Ref. 21].

- Presentation - concerned with data transformation, data formatting and data syntax. Allows an application to correctly interpret the data being transferred [Ref. 10].

- Session - provides the mechanism for controlling the dialogue between applications in end systems [Ref. 21].

- Transport - provides a reliable mechanism for the exchange of data between processes in different systems, it ensures that data units are delivered error-free, in sequence, with no losses or duplications [Ref. 21].

- Network - moves data through the network. carries out the functions of switching and routing, sequencing, logical channel control, flow control, and error recovery network wide [Ref. 10].

- Data link - provides services for reliable interchange of data across a data link established by the physical layer. The data link layer manages the establishment, maintenance, and the release of data link connections [Ref. 10].

- Physical - Provides the physical connectivity between two end users [Ref. 11].

In the OSI model all of the layers communicate with their peer through the physical layer. For example, assume two systems A and B both use the OSI model. If the two systems are communicating using an application level function then the systems will establish peer to peer relationships at that level [Ref. 21]. The relationship is conducted down through the layers of each individual system to the physical layer where the actual data bits are transferred, as depicted in Figure 2.1.

b. TCP/IP Architecture

The other major architecture is the Transfer Control Protocol/Internet Protocol (TCP/IP). TCP/IP is an outgrowth from development begun by ARPA for ARPANET and the Defense Data Network [Ref. 21]. TCP/IP has much more extensive implementation than OSI due primarily to its use by DoD [Ref. 20]. Like OSI, TCP/IP breaks up communication tasks into smaller more manageable subsections.

> The objection sometimes raised by the designers of the TCP/IP protocol suite and its protocols is that the OSI model is prescriptive rather than descriptive. It dictates that protocols within a given layer perform certain functions. This may not always be desirable. It is possible to define more than one protocol at a given layer, and the functionality of those protocols may not be the same or even similar [Ref. 21].

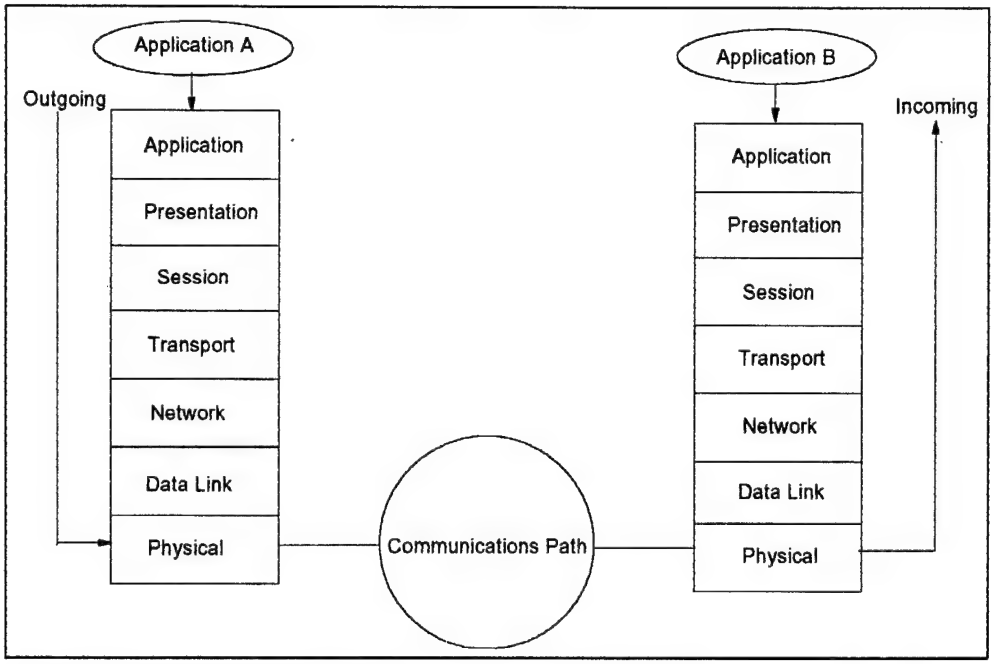


Figure 2.1 OSI architecture [Ref. 21]

TCP/IP relies on three factors to carry out successful communications: processes, hosts, and networks. Processes are the basic entities that communicate. The processes run on hosts. Communication between processes takes place over networks which contain hosts [Ref. 21].

The individual networks in a TCP/IP system are usually referred to as *subnetworks* [Ref. 21]. The IP portion of the protocol is implemented in both the individual systems or subnetworks and the routers, while TCP is implemented only on the individual systems. Every object must have a unique identifying address.

> Actually, two levels of addressing are needed. Each host on a subnetwork must have a unique global internet address; this allows the data to be delivered to the proper host. Each process with a host must have an address that is unique within the host; this allows the host-to-host protocol (TCP) to deliver data to the proper process. These later addresses are known as ports [Ref. 21].

TCP/IP is generally describe as having four layers: network access layer protocol (NAP), internet layer, host-to-host layer, and the process layer [Ref. 21]. The network access layer provides information which allows access to the network. The internet layer

allows communications to occur between multiple networks. The host-to-host allows process on different hosts to communicate. The process layer allows resource sharing and remote access [Ref. 21].

To look at a simple example , assume that a process is associated with host X port 1. X intends to give the message to a process on host Y port 2. The process on X sends the message to TCP with the instructions to send the message on to Y port 2. TCP then gives the message to IP with the instruction to send the message to host Y. IP is not concerned with the port address, only the host address. The message is then sent to the NAP, where a packet header is appended to the information. The packet header contains further instructions to be sent to the router. At the router the packet header is removed and the IP header is examined [Ref. 21]. The IP module in the router sends the datagram to Y. The message is again given NAP header information. When the data arrives at Y, information is stripped off in reverse until the destination is reached. An example of the TCP/IP architecture is depicted in Figure 2.2.

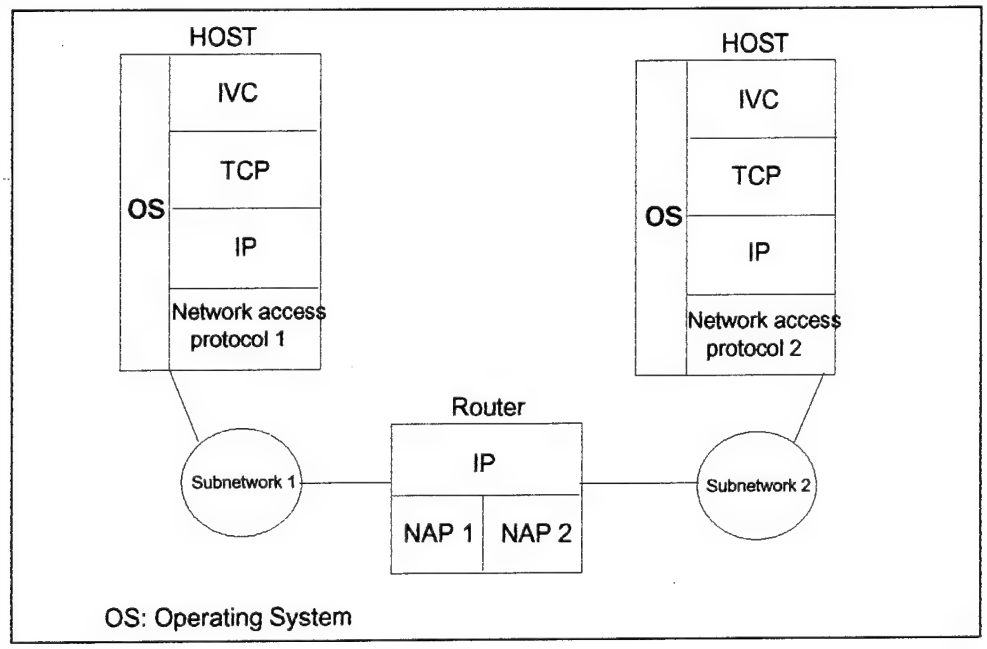


Figure 2.2 TCP/IP architecture [Ref. 21]

3. Network Encryption

Encryption in the network environment usually assumes one of two forms, link encryption, or end-to-end encryption. There are strong points and weak points for both. The decision on which form to use must be based on the security requirements of the organization and the information. Network information is inherently difficult to protect [Ref. 22]. An intruder can access information in a variety of ways. Network traffic analysis is very difficult to detect and even more difficult to prevent. Encryption is one way of protecting the network data.

a. Link Encryption

In link encryption all devices on the network are enabled with encryption capabilities. Devices connected at the physical layer encrypt all data passing through them, including data, routing information, protocol information, etc. The major strength of such a schema is that the information traveling over the network is extremely secure. Not only is the data itself encrypted, but the header information is encrypted as well. This points to one of the major drawbacks of link encryption, which is that the message must be decrypted whenever a packet reaches a switching point. The encrypted header and addressing information makes traffic flow analysis more difficult. Key management in link-to-link encryption can become a problem [Ref. 22]. Each device in the link must have the key for the data packet. This raises a serious problem with user authentication. A graphic depiction of link encryption is shown in Figure 2.3.

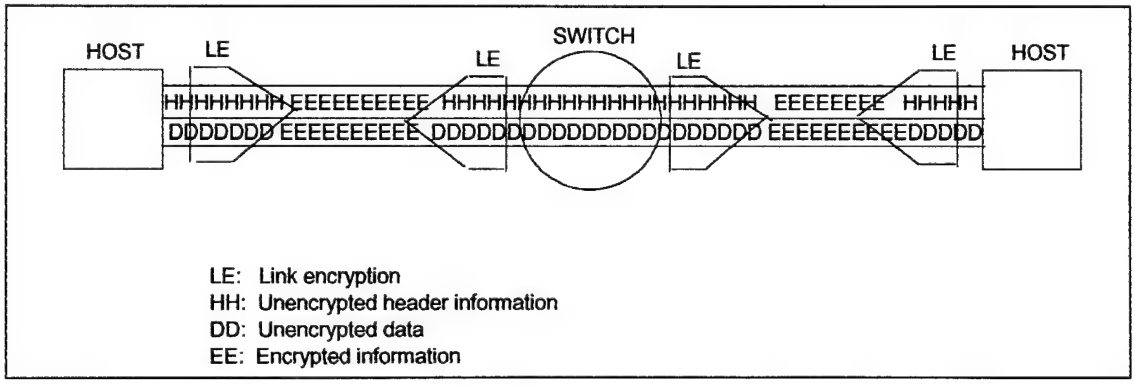


Figure 2.3 Link encryption [Ref. 22]

b. End-to-End Encryption

End-to-end encryption involves encrypting only the user data. The headers and addressing scheme are in the clear. The source user will encrypt the data. The encrypted data will be transmitted unchanged through the network to the destination. The end user will then decrypt the data. The obvious problem with this type of encryption is that there is no protection of the traffic flow information. One advantage that end-to-end encryption has, is that only the two end users share the key, reducing the authentication problem. In relating the end-to-end encryption to a communications architecture, end-to-end encryption occurs at higher levels. For instance in the OSI architecture, encryption may be placed at the application layer. A graphic depiction of end-to-end encryption is shown in Figure 2.4.

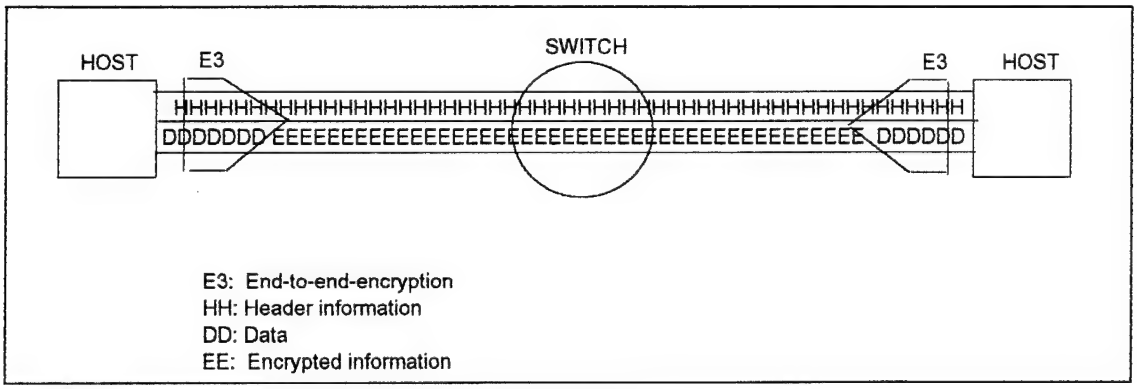


Figure 2.4 End-to-end encryption [Ref. 22]

E. SUMMARY

This chapter has presented an overview of the practice of and terminology associated with cryptography, as well as an overview of network practice and terminology. The chapter began by defining some of the basic terminology associated with networking and encryption. Next it described the basics of cryptography and gave a brief description of the three algorithms to be covered in this paper, DES, IDEA, and LOKI. Then it described the basics of networking and presented the two major network

architectures, TCP/IP and OSI. Finally, an introduction of the two major types of network encryption was presented.

III. DES, IDEA, AND LOKI CRYPTOSYSTEM DESCRIPTIONS

A. DES ALGORITHM

The Data Encryption Standard (DES) is a block cipher, symmetric cryptosystem incorporating both transposition and substitution. DES is a derivative of the LUCIFER cipher developed by IBM in the early 1970's [Ref. 19]. DES was adopted as a federal standard in 1976 and the algorithm was recertified for use in 1992. The algorithm uses a 64-bit key (only 56-bits are actually used). DES encrypts data in 64-bit blocks, taking a 64-bit block of plaintext and converting it into a 64-bit ciphertext block. There are two basic functions carried out in the DES algorithm, substitution and transposition. The algorithm uses only standard arithmetic and logical operations on numbers of at most 64-bits and is easily implemented in either hardware or software [Ref. 19]. There are four modes of operation for the DES specified in FIPS PUB 81: Cipher Block Chaining (CBC), Electronic Codebook (ECB), Output Feedback (OFB), and Cipher Feedback (CFB) [Ref. 15]. Because it is the most often used mode, this paper will detail the ECB mode of operation and describe the differences which occur in the other three modes of operation.

1. ECB Mode

There are sixteen rounds of operations performed on the text following an initial permutation. A "round" consists of one or more mathematical operations. These operations are repeated in successive "rounds" in any most cryptographic algorithms. An individual round of the DES is described in Figure 3.1. In the initial permutation:

> ... the input block is transposed as described in Table 3.1. This table ... should be read left to right, top to bottom. For example, the initial permutation transposes bit 1 to bit 58, bit 2 to bit 50, bit 3 to bit 42, etc. The initial permutation and the corresponding final permutation do not affect DES's security [Ref. 19].

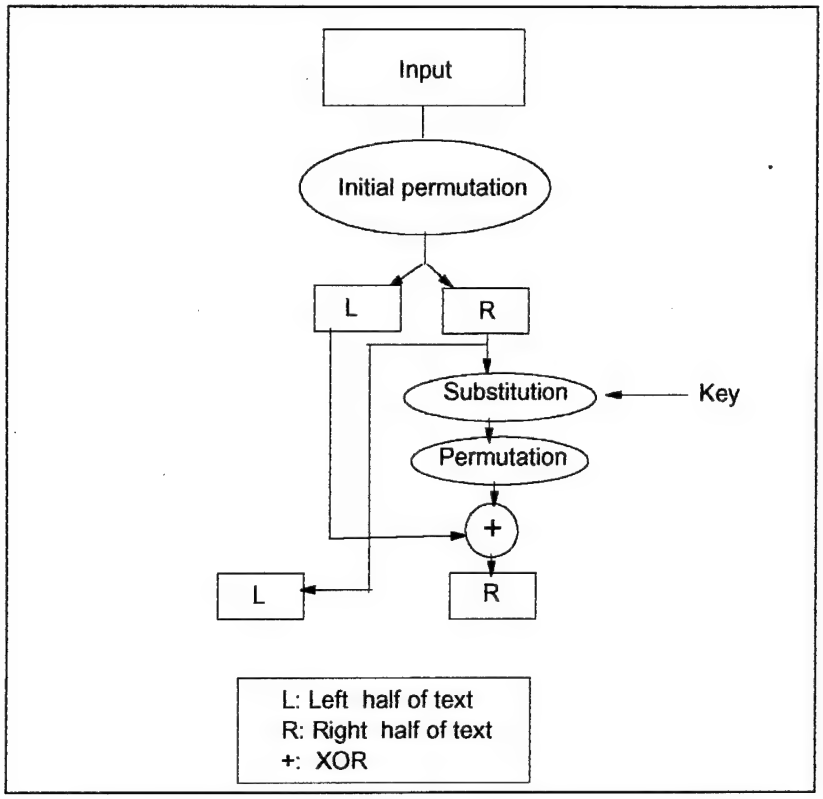


Figure 3.1 Initial Permutation and Round 1 of the DES [Ref. 19]

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Table 3.1 Initial Permutation [Ref. 19]

Following the initial permutation the 64-bit text is split into two 32-bit halves, a right and left half. The 32-bit right half is expanded into 48 bits using the expansion permutation. The expansion permutation is described in Table 3.2. The expansion permutation has two purposes: to make the intermediate halves of the ciphertext comparable in size to the key, and to provide a longer result that can be later compressed [Ref. 18]. Out of each 4-bit block the first and fourth bits are repeated, while the second and third are used only once.

Bit	1	2	3	4	5	6	7	8
Moves to	2,48	3	4	5,7	6,8	9	10	11,13
Bit	9	10	11	12	13	14	15	16
Moves to	12,14	15	16	17,19	18,20	21	22	23,25
Bit	17	18	19	20	21	22	23	24
Moves to	24,26	27	28	29,31	30,32	33	34	35,37
Bit	25	26	27	28	29	30	31	32
Moves to	36,38	39	40	41,43	42,44	45	46	47,1

Table 3.2 Expansion Permutation

The 64-bit key is reduced to a 56-bit key by disregarding every eighth bit. The DES uses a different 48-bit key for each round.

> First, the 56-bit key is divided into two 28-bit halves. Then the halves are shifted left by either one or two digits, depending on the round. After being shifted 48 out of the 56 bits are selected. Because this operation permutes the order of the bits as well as selecting a subset of bits, it is called a compression permutation, or the permuted choice. This operation provides a subset of bits the same size as the output of the expansion permutation [Ref. 19].

The key for the cycle is combined by an XOR function with the expanded right half of the input. The output is then sent into the S-boxes.

As stated earlier, one of the basic functions of the algorithm is substitution. This substitution occurs in the S-boxes. An S-box is a table by which six bits of data are replaced by four bits of data. The 48-bit output of the compressed key XORed with the expanded data block goes into the S-box. This output is split up into eight 6-bit blocks. There are eight total S-Boxes and each data block is operated on by a different S-box. The input bits determine the entry into the individual S-box.

> Suppose that block Bi is the six bits b1b2b3b4b5b6. Bits b1 and b6, taken together, form a 2-bit binary number b1b6, having a decimal value from 0 to 3. Call this value r. Bits b2, b3, b4, and b5 taken together form a 4-bit binary number b2b3b4b5, having a decimal value from 0 to 15. Call this value c. The substitutions from the S-boxes transform each 6-bit block Bi into a 4-bit result [Ref. 18].

The heart of the security of the DES rests with the S-boxes. The other operations in the DES are linear in nature, while the S-boxes are nonlinear. The results of this substitution phase are eight 4-bit blocks, which are recombined into a single 32-bit block [Ref. 19].

After the S-boxes the 32-bit data block is permuted with a straight permutation. This is done in the P-box. The P-box is described in Table 3.3. In the permutation process each bit of input is matched to an output position. The output of the P-box is XORed with the left half of the original 64-bit input. The halves are switched and the process begins again.

Bit	1	2	3	4	5	6	7	8
Moves to	2,48	3	4	5,7	6,8	9	10	11,13
Bit	9	10	11	12	13	14	15	16
Moves to	12,14	15	16	17,19	18,20	21	22	23,25
Bit	17	18	19	20	21	22	23	24
Moves to	24,26	27	28	29,31	30,32	33	34	35,37
Bit	25	26	27	28	29	30	31	32
Moves to	36,38	39	40	41,43	42,44	45	46	47,1

Table 3.3 Permutation Box

2. Cipher Block Chaining

The ECB mode encrypts each 64-bit block of data independently of other 64-bit block of data. Given the same key, identical plaintext will encrypt the same way. This greatly increases the likelihood of a successful plaintext attack. One way of overcoming this problem is to use the cipher block chaining mode.

In the cipher block chaining mode the plaintext is XORed with the previous cipher text before encryption. After encryption the resulting ciphertext is stored in a feedback register to be XORed with the next incoming text to the encryption routine.

Two messages that are the same will still encrypt to the same cipher text. This can be especially dangerous with message headers where the header information is always the same. In order to prevent this CBC uses an initializing vector (IV) on the first block of

data. The IV simply makes each message unique to reduce the likelihood of a plaintext attack.

3. Output and Cipher Feedback Modes

Both EBC and CBC modes require text to be encrypted in 64-bit blocks of plaintext. With output feedback (OFB) and cipher feedback (CFB) plaintext is encrypted in units smaller than the block size. For example, in 1-bit CBC data is encrypted 1 bit at a time.

> A block algorithm in n-bit CFB mode operates on a queue the size of the input block. Initially the queue is filled with an initializing vector as in CBC mode. The queue is encrypted, and the left-most n bits of the result are the XORed with the first n bit character of the plaintext to become the first n bit character of the ciphertext. The same n bits are also moved to the right-most n bit positions of the queue, and all the other bits move n to the left. The n left-most bits are discarded. Then the next character is encrypted in the same manner [Ref. 19].

In the OFB mode part of the previous output is put into the right-most positions in the queue. Both the OFB an CFB modes use an initializing vector.

4. Cryptanalysis of the DES

The security of the DES has been an issue of concern since it was first established as federal standard in 1976. Of primary concern is the key length. The original Lucifer cipher supported a 112-bit key. Many cryptographers feel that the National Security Agency (NSA) and the National Bureau of Standards (NBS), now National Institute of Standards and Technology (NIST), had a hand in reducing the key length to 56 bits and installing a "trapdoor" [Ref. 19]. Regardless of NSA's input into the development of the DES, the algorithm has proven itself to be a very secure symmetric key algorithm. There are several weaknesses which could be exploited.

The ever increasing power and speed of processors makes the likelihood of an exhaustive key search a possibility. With a 56-bit key there are 2^{56} possible keys to try. It is estimated that a massively parallel system could do this key search in 1 day, however the cost of such an operation would be tremendous [Ref. 7].

Because of the way the key is used at the beginning of each round some keys will remain the same throughout the encryption cycle. These keys are called weak keys. If the key consists of all 0's or all 1's or if one half consists of entirely 1's and the other half entirely 0's or the other way around, then the result is a weak key. There are also pairs of keys which will encrypt plaintext into the same ciphertext. These pairs of keys are called semi-weak keys. There are a total of 64 weak or semi-weak keys out of a set of 72,057,594,037,927,936 possible keys [Ref. 19]. This is a relatively small and avoidable subset of the total keyspace.

B. IDEA ALGORITHM

The International Data Encryption Algorithm (IDEA) is a block cipher, symmetric cryptosystem incorporating both confusion and substitution. The algorithm was introduced by Xuejia Lai and James Massey in 1990. The algorithm uses a 128-bit key. IDEA encrypts data in 64-bit blocks, taking a 64-bit block of plaintext, breaking it up into 4, 16-bit blocks to employ the algorithm. The design philosophy behind the algorithm is one of mixing operations from different algebraic groups: XOR, addition modulo 2^{16}, and multiplication modulo $2^{16}+1$ [Ref. 19]. These operations add confusion to the algorithm. There are eight rounds in the algorithm. IDEA operates in the same four modes as DES, electronic codebook, cipher block chaining, cipher feedback, and output feedback modes. The modes carry out the same function in IDEA as they do in DES. IDEA has achieved wide recognition as an extremely capable cryptosystem.

> IDEA is one of a number of conventional encryption algorithms that have been proposed in recent years to replace DES....In terms of adoption, IDEA is by far the most successful of these proposals. For example, IDEA is included in PGP [Pretty Good Privacy cryptosystem developed by Phil Zimmerman], which alone assures widespread use of the algorithm [Ref. 22].

1. Function

Initially a 64-bit block of plaintext is taken and divided into four 16-bit sub-blocks. The 128-bit key is also split into a total of fifty-two subkeys. Generation of the subkeys is

a relatively simple process [Ref. 19]. The first six subkeys are taken directly from the key, with subkey 1 being the 16 most significant bits, subkey 2 being the 16 next most significant bits, until subkey 6.

> ...a circular left shift of 25 bit positions is applied to the key, and the next eight subkeys are extracted. This procedure is repeated until all 52 subkeys are generated....This scheme provides an effective technique for varying the key bits used for subkeys in the eight iterations. Note that the first subkey used in each round uses a different set of bits from the key [Ref. 22].

Each of the eight rounds uses six 16-bit subkeys. The four 16-bit sub-blocks of text are combined with four subkeys using addition and multiplication, producing four output blocks. This is the initial transform. Between each round, the second and third sub-blocks are swapped [Ref. 19]. The four output blocks of text are XORed to form two 16-bit blocks of text. The two blocks of text are then input into the multiplication/addition (mult/add) box. The mult/add box adds diffusion to the algorithm.

> The multiplication/addition box is the basic building block of the algorithm. This structure takes as inputs two 16-bit values derived from the plaintext and two 16-bit sub-keys derived from the key and produces two 16-bit outputs. An exhaustive computer check has determined that each output bit of the first round depends on every bit of the plaintext-derived inputs and on every bit of the sub-keys. This particular structure is repeated eight times in the algorithm, providing very effective diffusion. Furthermore, it can be shown that this structure uses the least number of operations (four) required to achieve complete diffusion [Ref. 22].

The two output blocks of the mult/add box are XORed with the four output blocks of the initial transform. The second and third output blocks are swapped. This increases the mixing of the bits being processed and makes the algorithm more resistant to differential cryptanalysis. These four output blocks serve as the input to the next round.

All of the rounds carry out the same functions, however after the eighth round there is another transform carried out. The transform is similar to the initial transform in the beginning of the algorithm. The second and third sub-blocks of the output from round

eight are swapped again. The four sub-blocks are then added and multiplied, just as in the initial transform, to produce four sub-blocks. These four sub-blocks are then put together to form the ciphertext. An individual round of IDEA is described in Figure 3.2.

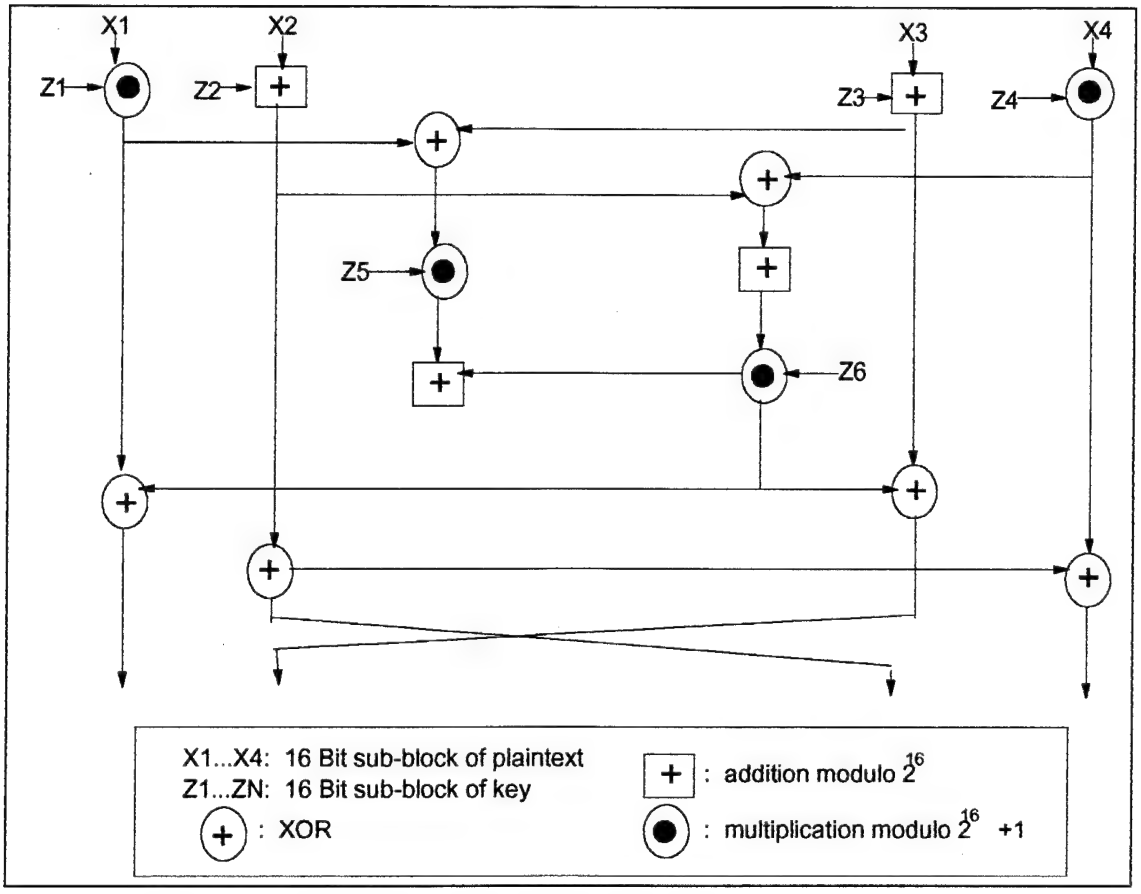


Figure 3.2 An individual round of IDEA [Ref. 22]

2. Cryptanalysis of IDEA

IDEA is still a new algorithm. There have been no papers published on the cryptanalysis of IDEA [Ref. 19]. There are elements of the algorithm which give it cryptographic strength. The real strength of the algorithm lies in the key size. With a key length of 128-bits an exhaustive key search is out of the question with today's technology. On a computer capable of trying a million keys a second, it would take 10^{25} years to find

the key [Ref. 19]. Numerous organizations are certainly cryptanalyzing IDEA. It currently appears to be secure and appears to be more secure than DES [Ref. 19].

C. LOKI ALGORITHM

The LOKI algorithm is a block cipher, symmetric cryptosystem which operates very similarly to DES. LOKI is an Australian cryptosystem, first presented in 1990. Several problems with the security of the algorithm were discovered and the algorithm was redesigned. The initial algorithm was renamed LOKI89, and the new algorithm was named LOKI91. LOKI in this paper refers to the LOKI91 algorithm. LOKI encrypts data in 64-bit blocks, taking a 64-bit block of plaintext, breaking it up into two 32 bit blocks to employ the algorithm. LOKI uses many of the same arithmetic and logical operators as DES. Unlike DES there is no initial Permutation. The algorithm uses a 64-bit key.

1. Function

Initially a 64-bit block of plaintext is taken and divided into two 32-bit sub-blocks. The 64-bit key is also split into two 32-bit subkeys.

> The key schedule is responsible for deriving the subkeys. In each round i, the subkey Ki is the current left half of the key . On odd numbered rounds this half is then rotated 12 bits to the left. On even numbered rounds, this half is then rotated 13 bits to the left, and the key halves are interchanged [Ref. 12].

In each round there are three operations which occur: expansion permutation, S-box, which has the same function as the S-box in the DES,, and straight permutation. These three operations are called the encryption function. The expansion permutation takes a 32-bit input and produces a 48-bit output. The output from the expansion permutation is divided into four sub-blocks of 12 bits each and serves as the input into the four S-boxes.

The S-boxes in the LOKI algorithm perform the same function as the S-boxes in the DES. The S-boxes add confusion to the cipher. The S-boxes take the 12-bit output from the expansion permutation and produce an 8-bit output. The 8-bit output are concatenated together to form the 32-bit output of the S-boxes. The 8-bit output from S-box(4)

becomes the most significant byte (bits [31...24]), then the outputs from S-box(3) (bits[23...16]), S-box(2) (bits[15...8]), S-box(1) (bits[7...0]) [Ref. 12].

The output of the S-boxes is sent to the permutation function. Permutation adds diffusion to the algorithm.

> The permutation function takes the 32-bit concatenated outputs from the S-boxes, and distributes them over all the inputs for the next round via a regular wire crossing which takes bits from the outputs of each S-box in turn [Ref. 12].

The 32-bit output of the encryption function is then added modulo 2 to the left 32-bit block of input text. The two input block halves are then swapped. This process is repeated for 16 rounds except that the input block halves are not swapped in the last round. After the last round the output blocks are concatenated together to form the output block [Ref. 12]. The first two rounds of LOKI are described in Figure 3.3.

2. Cryptanalysis of LOKI

Of the three algorithms reviewed in this paper LOKI is probably the weakest cryptographically. The redesign of LOKI89 did enhance the algorithm significantly, however there are still several weaknesses. The original algorithm was susceptible to differential cryptanalysis with eleven or fewer rounds [Ref. 2]. The algorithm also suffered from a key complementation problem which reduces the exhaustive key search by a factor of 256 [104,512,513][Ref 19]. LOKI91 was redesigned with the following changes:

- The key schedule was changed to swap halves after every second round [Ref. 12].
- The subkey generation algorithm was changed so that the rotation of the left subkey alternated between 12 and 13 bits to the left [Ref. 19].
- The initial and final XOR of the block with the key was eliminated [Ref. 19].
- The S-box function was altered [Ref. 12].

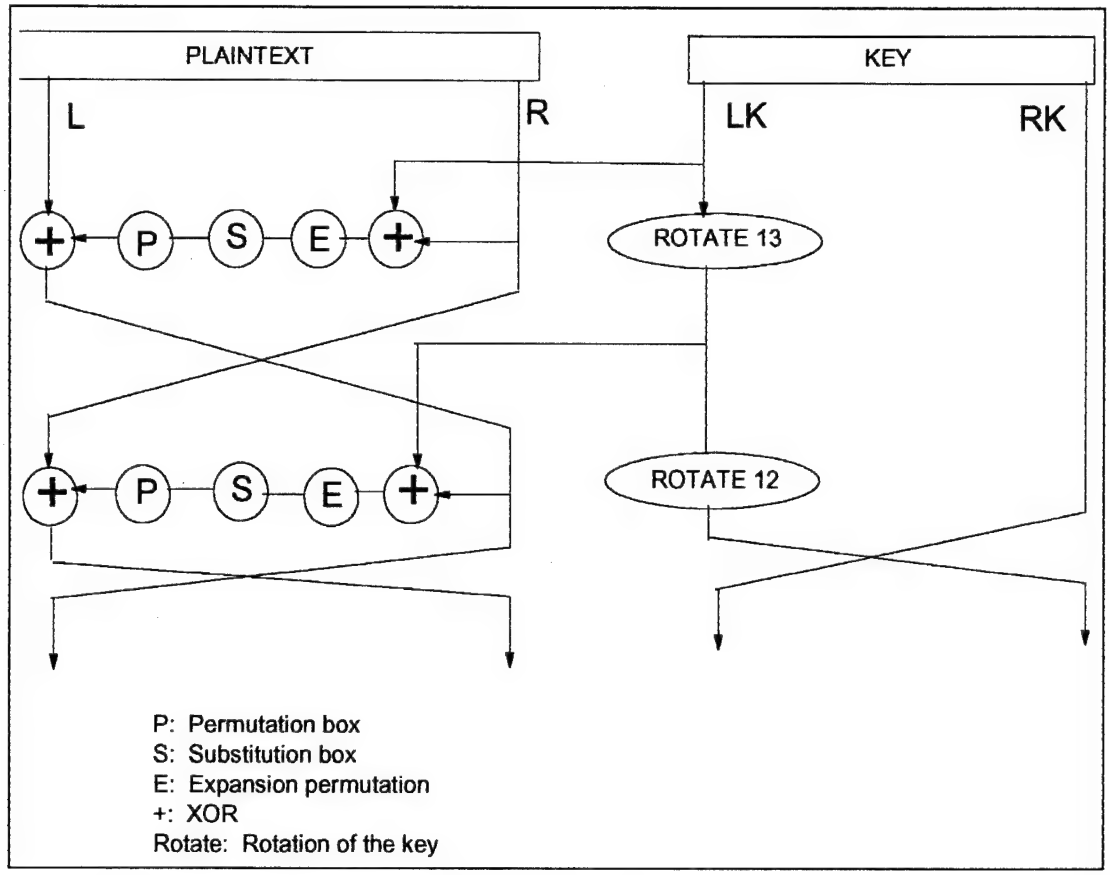


Figure 3.3 First two rounds of LOKI [Ref. 19]

The redesigned output should be immune to differential cryptanalysis [Ref. 12]. The exhaustive key search is reduced by a factor of four, due to a weakness in the key schedule [Ref. 19]. There is a new version of LOKI currently being developed which should eliminate some of the current weaknesses. At present LOKI cannot be considered a secure cipher.

D. SUMMARY

All three of the cryptosystems presented here, DES, IDEA, and LOKI, have their own individual strengths and weaknesses. This chapter has presented a detailed analysis of all three cryptosystems. Table 3.4 gives a representation of some features of the major characteristics of each of the three cryptosystems covered in this chapter.

	IDEA	DES	LOKI
KEY LENGTH	128 Bits	56 Bits	64 Bits
BLOCK SIZE	64 Bits	64 Bits	64 Bits
ROUNDS	9	16	16
SUB-BLOCK SIZE	16 Bits	32 Bits	32 Bits
VULNERABILITY	LOW	LOW	HIGH

Table 3.4 Cryptosystem vulnerabilities

IV. AN IMPLEMENTATION PROPOSAL

A. INTRODUCTION

Encrypted voice communication in computer networks involves four phases. The analog voice signal is digitized. The digitized voice data is then encrypted and put into packets. The packets are then transmitted over the network. The packets are received and decrypted at the other end [Ref. 6].

In order to establish a plan for implementing encrypted voice traffic over a PC network, several factors must be considered. Some of the primary factors have been discussed in this paper. The main factors are: whether to use link or end-to-end encryption; the architecture to base the design on, OSI or TCP/IP; and the encryption algorithm to be used. The focus of this chapter will be to make recommendations on these key factors, which should assist in the implementation of encrypted voice traffic in a PC network. Several applications exist both in shareware software and commercial software which can aid in the implementation.

B. ENCRYPTION METHODOLOGY

End-to-end encryption is the recommendation for the encryption methodology. End-to-end encryption relieves some of the key management problem which would become extremely troublesome in link encryption. The encryption scheme can be implemented directly at the application layer making it easier to implement in software, and making it independent of the type of communication network used [Ref. 19]. The user information remains encrypted from point of origin to point of destination. Link encryption decrypts the entire message at each node making the information vulnerable at these points.

C. ARCHITECTURE

The architecture should utilize the Transmission Control Protocol/Internet Protocol Architecture (TCP/IP). The architecture provides a way for the applications to communicate reliably between two computers. As described in Chapter II, TCP/IP

incorporates four layers: an application protocol, the transport protocol, the internet protocol, and the lower level protocol needed to manage the physical link. The user should be able to access another computer on the network simply by knowing another IP address. The routing should be invisible to the user.

TCP/IP is a "connectionless" technology. The packet information is transferred as a sequence of "datagrams." Provisions are made to open connections (conversations which will continue for some time). The datagrams are eventually broken up and treated as separate entities. So for example if the voice transmission is a 25,000 byte file, the TCP/IP protocol may break it up into 50, 500 byte datagrams. The datagrams will be transmitted as separate entities through the network and will be put back together at the other end.

The recommendation at the physical layer is for an Ethernet implementation. Ethernet is an operational local network designed to provide cost-effective interconnection for a large number of heterogeneous data users. At the data link level data packets (or frames) are transmitted over a single cable by means of a multi-access contention mechanism [Ref. 6].

D. NETWORK VOICE APPLICATION

A shareware product called "Internet VoiceChat" (IVC), handles internet or network voice traffic. IVC was created by Richard Ahrens at the University of Pennsylvania Wharton School [Ref. 1]. The IVC application is useful because it handles all networking tasks of the voice traffic. IVC requires the following: Winsock 1.1, a Windows™ compatible sound card with microphone, net connectivity (e.g., Ethernet), and a 386 processor or better CPU. The total size of the installed software is 189,322 bytes. The source code for the product is reportedly available for academic institutions to use for academic research.

1. Installation

Installing the IVC product is a relatively easy process. After running the installation program, a setup screen is presented as displayed in Figure 4.1. Several of the items in the setup screen are not available in the current version of the software. The "Call Screen Timeout" window sets the interval the program will wait before it defaults to the specified option when the "Call Screen" is active. The two "Call Screen" modes are: "answering machine," which records a message left by another user; and "reject," which forces the call to be disconnected.

Two directories are specified in the setup menu. The first is the "message directory", which dictates the disk location to store the recorded messages. The second directory is the "working directory", for the IVC files. It is recommended by the vendor that the "working directory" be a RAM disk with 700k free space [Ref. 1].

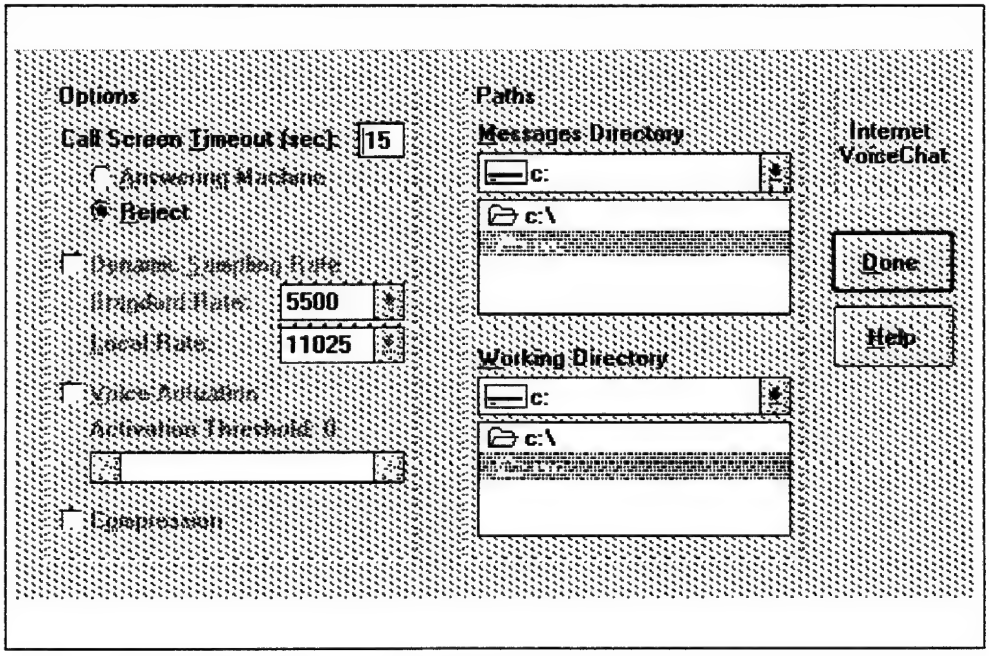


Figure 4.1 IVC setup screen [Ref. 1]

2. Use

To initiate a call, "VoiceChat" is selected in the "Mode" box. The remote IP address of the individual to be called is entered into the "Dial" field. Once the call button is pressed the menu will indicate that the user is attempting to be contacted. After the two parties are connected the record and stop buttons on the menu will light up. The program uses a traffic light symbol to indicated when the user can record and transmit. The program will allow for up to one minute of recorded data. When the user has finished recording the stop button is pressed. The traffic light symbol will turn red and the message will be transmitted. Once the traffic light turns green again, the other user may respond. The traffic light symbol will turn red when the user is recording or sending her or his voice. A depiction of the call screen is presented in Figure 4.2.

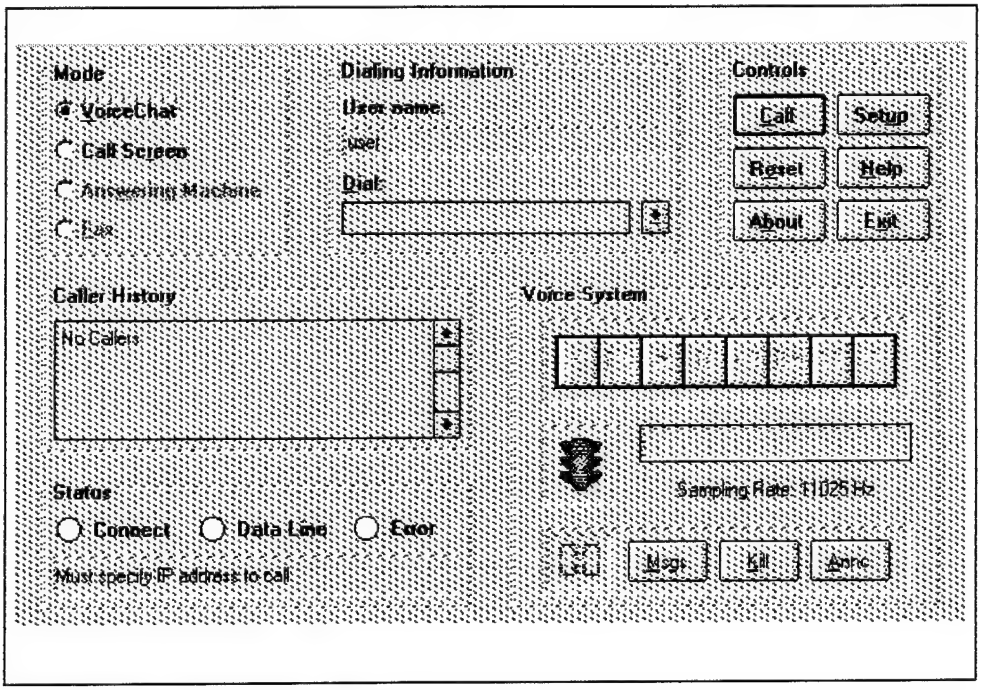


Figure 4.2 IVC call screen [Ref. 1]

E. ENCRYPTION ALGORITHM

Using the IVC source code it is possible to include an encrypting function within the IVC application. Source code is readily available for all three of the cryptographic

algorithms reviewed in this paper. Because of the lack of cryptographic security provided by the LOKI algorithm it should not be considered as a choice. The DES and IDEA algorithms both provide proven security.

1. DES

Several shareware programs exist which allow for the implementation of the DES. A shareware program called *The Private Line* written by Everett Enterprises provides a useful means for testing the algorithm [Ref. 8]. The DES source code is available for academic users. The program conforms to and demonstrates all 171 of the tests required by NIST for a full DES implementation [Ref. 8]. An example screen of one of the tests required by NIST is depicted in Figure 4.3.

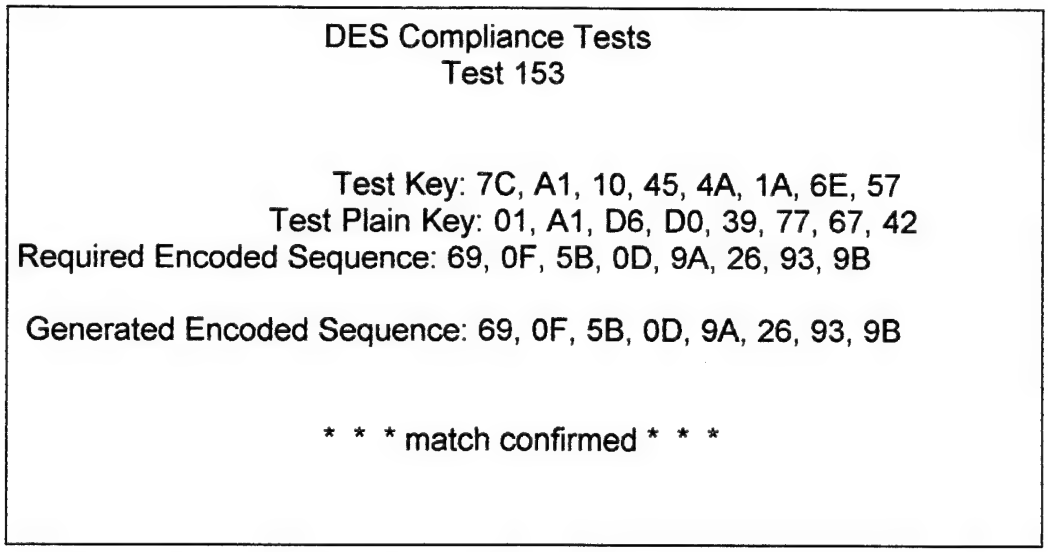


Figure 4.3 NIST test 153 of the DES [Ref. 8]

As an example of the actual encryption, a sample file consisting of the letters "abcdef", 61 62 63 64 65 66 0D 0A in hexadecimal format, was encrypted using the DES code. The encrypted file in hexadecimal is 86 6C E4 A2 F2 35 02 36.

2. IDEA

Several shareware programs exist which allow for the implementation of IDEA. A shareware program called *MASK* written by Eugenio Ciruana and Dominick Pallone,

provides a useful means for testing the algorithm [Ref. 17]. The IDEA and *MASK* source codes are available for academic and non-commercial use. The IDEA algorithm operates on 16-bit blocks and is very efficient in software implementations even on 16-bit processors [Ref. 19]. As an example of the actual encryption, an sample file consisting of the letters "abcdef", 61 62 63 64 65 66 0D 0A in hexadecimal format, was encrypted using the IDEA code. The encrypted file in hexadecimal is 65 A8 3A 5C 37 C1 CC 40 20 70 08 00 00 00 08 00 00 00 7A 53 7A 17 86 84 93 59 00 20 51 C0 00 88.

3. Experimental results

Several experiments were performed using the *Mask* product and the *Private Line* product as a means of comparing the speed and efficiency of the DES and IDEA algorithms. In all of the experiments, "wall time" was used in order to measure the speed. The programs were run on a 386/33mHz IBM compatible PC.

The first set of experiments were designed to determine if there was any "overhead" or excess data in the encrypted text associated with either of the algorithms. Small test files from 32 bits up to 128 bits were used to test for "overhead." Both algorithms were run in the CBC mode. The results of the these tests are displayed in Table 4.1.

Algorithm	Plaintext	Ciphertext
DES	4 Bytes	N/A
DES	8 Bytes	8 Bytes
DES	12 Bytes	12 Bytes
DES	16 Bytes	16 Bytes
IDEA	4 Bytes	26 Bytes
IDEA	8 Bytes	26 Bytes
IDEA	12 Bytes	34 Bytes
IDEA	16 Bytes	34 Bytes

Table 4.1 Plaintext file size versus encrypted file size

The Private Line DES implementation will not encrypt a file smaller than 64 bits in size unless the padding option is implemented. There are two ways to deal with plaintext that does not match the 64-bit block size. The easiest way to match the block size is to "pad" the input. Padding adds data to make the input block equal a 64-bit block size. The other option is to encrypt the last short block size differently then the rest of the blocks. This is more a complicated option and can make the algorithm much less efficient [Ref. 19]. The *MASK* IDEA implementation will encrypt and decrypt files smaller than 64 bits. The *MASK* program adds a marker to the ciphertext to identify the key used to encrypt. The 18-byte marker serves to prevent a file from being encrypted again by a different user. The results of the encryption show that both algorithms encrypt without adding significant overhead. The extra file length in the IDEA algorithm is due to the marker and "padding:"

Several audio files of differing sizes and time lengths were recorded, encrypted, and decrypted to test the speed of the algorithms. The files were recorded using a Sound Blaster™ Multi-CD 16 sound card. The sound card was set with 8-bit Mono, 11,025Hz sampling, PCM (*Microsoft's* uncompressed format) settings. These are the same settings the IVC product uses for its file recordings. Audio file samples of 1 second, 10 seconds, 20 seconds, 30 seconds, and 1 minute were taken. The results of the encryption time tests are displayed in Figure 4.4. The results of the decryption time tests are displayed in Figure 4.5. DES is a slightly faster algorithm for encrypting the files. At the 1 minute file size, the maximum file length for the IVC product, DES is slightly more than 7 seconds faster than the IDEA algorithm. Both algorithms follow a relatively steady curve as the file size increases.

The decryption times are only slightly different than the encryption time. Allowing for timing error, the times are virtually the same. All of the files were tested for their ability to give a playback after decryption. All of the files played successfully.

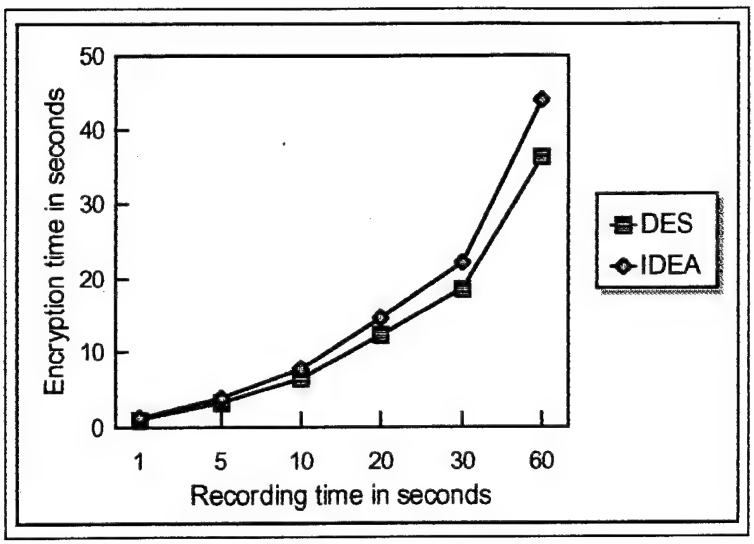


Figure 4.4 DES and IDEA encryption times

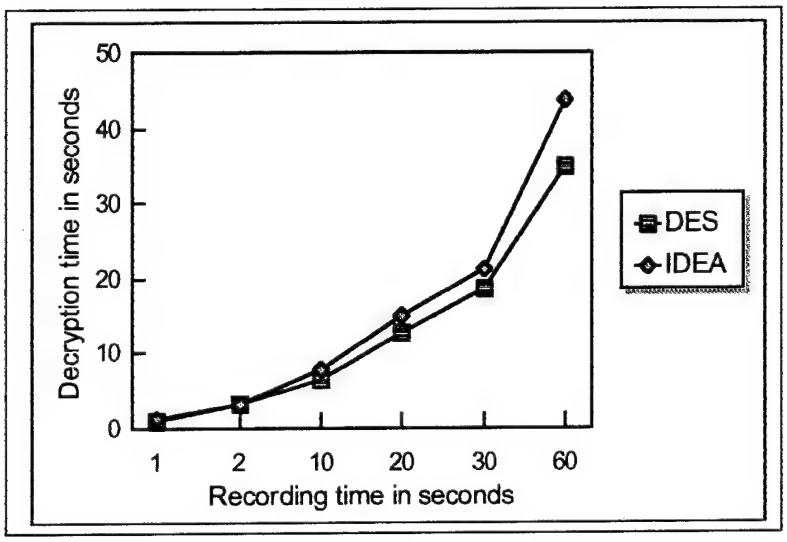


Figure 4.5 DES and IDEA decryption time tests

The algorithms are very similar in speed and in the level of complexity for implementation. The greatest difference between the two algorithms is the level of security. IDEA is considered to be a much better algorithm than the DES [Ref. 19]. The IDEA algorithm should be used in the encrypted voice network implementation because of its security advantage.

F. SUMMARY

This chapter has provided a proposal for implementing encrypted voice in a PC network. The chapter began by presenting some of the key factors which must be considered in an encrypted voice network implementation. The results of experiments done with the DES and IDEA encryption algorithms were presented to compare the capabilities of the two algorithms.

A detailed presentation of the Internet Voice Chat software program was presented as a possible solution to one of the problems faced in implementing encrypted network voice. Additional recommendations, and justifications were made to assist follow on research in a future implementation.

V. CONCLUSIONS

A. OVERVIEW

Protecting the information on expanding computer networks is of vital concern to DoD. As DoD and DoN move to expand the use of the available bandwidth and put more and more applications onto networks, security will become a continually growing problem. The Computer Security Act of 1987 mandates that all government agencies protect sensitive information which is subject to the Privacy Act if it is stored or transmitted on a computer [Ref. 23].

The implementation of voice into the networking arena brings even more challenges in security. The level of security in these networks must be of utmost concern. Encryption is just one possible means of protecting the information on computers and computer networks. Numerous encryption algorithms have been developed with varying levels of security. In selecting an algorithm, level of security and ease of implementation should be of prime importance.

B. REVIEW OF RESEARCH QUESTIONS

1. Primary Research Question

What are the capabilities, effectiveness, and limitations of LOKI, Data Encryption Standard (DES) and International Data Encryption Algorithm (IDEA) cryptosystems? All three of the algorithms have relative strengths and weaknesses which set them apart from each other. The LOKI algorithm has several weakness involving the security of the algorithm whch make it unsuitable for use. DES and IDEA both have ease of implementation and strong levels of security in their favor. Key length is a primary factor in the security of these cryptosystems. The IDEA algorithm uses a 128-bit key as opposed to the 56-bit key length for the DES. The longer key length provides for greater security. Both algorithms are relatively easy to implement in the PC environment.

2. Secondary Research Question 1

What are the major areas of concern with encryption in network configuration? One of the first issues to be addressed in encrypted network configuration is the type of architecture to use in the network. The two major types of computer network architecture are TCP\IP and OSI. The TCP\IP architecture suite is the most commonly implemented architecture and is supported by all of the software products mentioned in this thesis. The OSI model is a standard model however it is not as widely implemented as TCP\IP. The other key factor to consider is whether to use link or end-to-end encryption. Both methods have their own advantages and disadvantages however end-to-end encryption provides the most reasonable solution for this thesis. End-to-end reduces the key management burden and allows an easier software solution than does link encryption.

3. Secondary Research Question 2

What are the possibilities for implementing an encrypted PC voice network in a Naval Postgraduate School? While this thesis does not directly address an implementation scheme, several recommendations were made which will aid in the implementation of an encrypted voice PC network at the Naval Postgraduate School. The source code for the IVC product was not obtained in time to integrate the encryption source code into it. The level of programming required for such an integration was also beyond the scope of this thesis.

Implementing an encrypted voice network is certainly possible. Several products were mentioned which would enable such an implementation. Future research should be focused toward the programming aspect of the implementation. Strong knowledge of the C++ programming language is essential for future work. Future research should also extend beyond the confines of the PC network. The ideas presented here are certainly portable to other platforms and networks, such as Apple™ and UNIX.

APPENDIX: GLOSSARY OF TERMS

Computer Security Act of 1987: Under the Computer Security Act the National Institute of Standards and Technology (NIST) was given responsibility for improving computer security in civilian government agencies. This responsibility was taken away from the National Security Agency (NSA). Under the new law, NIST must create security standards, design security measures, and develop training programs for computer security [Ref 18].

Confusion: Obscures the relationship between the ciphertext and the plaintext Ref [19].

Connectionless: A service in which data are transmitted from one entity to another without the prior mutual construction of a connection (e.g., datagrams) [Ref 21].

Cryptanalysis: The science of recovering a plaintext message from the ciphertext without the key [Ref 19].

Datagram: A self-contained packet that carries information sufficient for routing from the originating data terminal equipment [Ref 21].

Differential Cryptanalysis: A cryptoanalytic tool which looks at pairs of plaintexts and determines the probabilities of reappearing in the resulting ciphertext to determine the most likely key.

Diffusion: Dissipates the redundancy of the plaintext by spreading it out over the ciphetext.

High-Level Data Link Control (HDLC): A very common bit-oriented data link protocol (OSI layer 2) issued by ISO. Similar protocols are ADCCP, LAP-B, and SDLC [Ref 21].

International Organization for Standardization: An international agency for the development of standards on a wide range of subjects. It is a voluntary, nontreaty organization whose members are designated standards bodies of participating nations [Ref 21].

Internetworking: Communication among devices across multiple networks [Ref 21].

National Institute of Standards and Technology (NIST): Part of the Department of Commerce, it issues Federal Information Processing Standards (FIPS) for equipment sold to the federal government.

National Security Agency (NSA): NSA has the authority to develop a national policy on communications and computer security which extends to the private sector in cases where the implementation of security would affect the national interests [Ref 18].

Network: Communication among a variety of data communicating devices within a small area [Ref 21].

Packet: Data or a portion of data which also contains control information (packet header) that gives the destination [Ref 21]

Router: Routes packets between potentially different networks [Ref 21].

Shareware: Software which allows a potential user to try out a package before buying it. The authors encourage free sharing of the software [Ref 18.]

Stovepipe System: A system developed without regard to generally accepted standards for use by a single department or agency.

Traffic Analysis: A tool used to determine the location and identity of communicating hosts. It is also used to observe the frequency and length of messages being exchanged [Ref 22].

Wall Time: A timing method for a process, function or software which relies on a standard clock rather than an internal timing mechanism.

LIST OF REFERENCES

1. Ahrens, R. *Internet VoiceChat Version1.0 Help Manual.* Pennsylvania: University of Pennsylvania, 1994.

2. Biham, E. and Shamir, A. "Differential Cryptanalysis of Snefru, Khafre, REDOC-II, LOKI, and Lucifer," *Advances in Cryptology-CRYPTO '91 Proceedings.* Berlin: Springer-Verlag, 1992.

3. Black, U. *Emerging Communications Technology.* Englewood Cliffs, New Jersey: Prentice Hall, 1994.

4. Brown, L. *Analysis of the DES and Its Implications for the Design of an Extended DES.* Canberra: Australian Defence Force Academy, 1993.

5. Brown, L. and Gilje, M. *Secure File Transfer Over TCP/IP.* Canberra: Australian Defence Force Academy, 1992.

6. Chlamtac, I. "An Ethernet Compatible Protocol for Real-Time Voice/Data Integration," *Perspectives on Packetized Voice and Data Communications.* New York: IEEE Press, 1991.

7. Computer Science Department. *Course Notes for CS4601: Computer Security.* Monterey, CA: Computer Science Department, 1993.

8. Everett, S. *Implementation of Private Line.* Springfield, VA: Everett Enterprises, 1990.

9. Fahn, P. *Answers to Frequently Asked Questions About Today's Cryptography.* Redwood City, CA: RSA Laboratories, 1993.

10. Freeman, R. *Telecommunication Transmission Handbook.* New York: John Wiley & Sons, Inc., 1991.

11. Herrtwich, R. (Ed.) *Network and Operating System Support for Digital Audio and Video.* New York: Springer-Verlag, 1992.

12. Kwan, M. and Seberry, J. *Improving Resistance to differential Cryptanalysis and the Redesign of LOKI.* Canberra: Australian Defence Force Academy, 1991.

13. Lai, X. and Massey, J. "A Proposal for a New Block Encryption Standard." *Advances in Cryptology-EUROCRYPT '90 Proceedings.* Berlin: Springer-Verlag, 1990.

14. National Institute of Standards and Technology. *Federal Information Processing Standards Publication 140-1: Announcing the Standard for Security Requirements for Cryptographic Modules.* United States Government Printing Office, 1994.

15. National Institute of Standards and Technology. *Federal Information Processing Standards Publication 81: DES Modes of Operation.* United States Government Printing Office, 1994.

16. Pallone, D. and Eugenio Ciurana. *MASK 1.0 User's Guide.* San Francisco: CIME Software Laboratory, 1994.

17. Pallone, D. and Eugenio Ciurana. *MASK 1.0 Technical Referance Guide.* San Francisco: CIME Software Laboratory, 1994.

18. Pfleeger, C. *Security in Computing.* Englewood Cliffs, New Jersey: Prentice Hall, 1989.

19. Schneier, B. *Applied Cryptography.* New York: John Wiley & Sons, Inc., 1994.

20. Sprague, R. and McNurlin B. *Information Systems Management in Practice.* New Jersey: Prentice-Hall, 1993.

21. Stallings, W. *Data and Computer Communications.* New York: Macmillan, 1994.

22. Stallings, W. *Network and Internetwork Security: Principles and Practice.* New Jersey: Prentice Hall, 1995.

23. United States Government. *Public Law (100-235) Computer Security Act of 1987.* Unites States Government Printing Office, 1987.

INITIAL DISTRIBUTION LIST

	Number of Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2. Library, Code 52 Naval Postgraduate School Monterey, California 93943-5101	2
3. Chin-Hwa Lee, Code EC/LE ECE Department Naval Postgraduate School Monterey, California 93943-5101	2
4. Myung W. Suh, Code SM/SU Department of Systems Management Naval Postgraduate School Monterey, California 93943-5101	2
5. Lieutenant Walter O. McClenney Post Office Box 87 Lawrenceville, Virginia 23868	2